

CLAIMS

What is claimed is:

1. A method of generating a device driver having a plurality of functional components, the method comprising:
 - generating a file for each functional component of the device driver and a user interface;
 - rebuilding only a file corresponding to the functional component to be modified to make a device driver installation program when a modification of a functional component is needed; and
 - reconstructing the device driver information using a file corresponding to the functional component selected by a user on the device driver installation program and generating an update to the device driver using the reconstructed device driver information.
2. The method of claim 1, wherein the generating the file comprises constructing each functional component according to a model and functions of a device and generating the file for each functional component.
3. The method of claim 1, wherein the device driver is used in a printer, and each functional component has a data structure value of a DEVICE MODE (DEVMODE) as a parameter.
4. An apparatus generating a device driver having a plurality of functional components, the apparatus comprising:
 - an installation program maker which generates a file for each functional component of the device driver and a user interface and rebuilds only the file corresponding to a functional component to be modified to make a device driver installation program; and
 - a driver generator which reconstructs information on the device driver to be installed using the file corresponding to the functional component selected by a user on the device driver installation program and generates an update to the device driver using the reconstructed device driver information.

5. The apparatus of claim 4, wherein the device driver is used in a printer, and each functional component has a data structure value of a DEvIce MODE (DEVMODE) as a parameter.

6. A method of generating a user interface screen for a device driver, the method comprising:

determining whether a particular information file including menu generation information for the user interface screen depending on a device model exists;

when the particular information file exists, extracting the menu generation information from the particular information file; and

generating the user interface screen for the device driver based on the extracted menu generation information.

7. The method of claim 6, wherein the determining is performed when opening the user interface screen.

8. The method of claim 6, further comprising:

generating the user interface screen based on predetermined default values when the particular information file does not exist.

9. An apparatus generating a user interface screen for a device driver, the apparatus comprising:

a file detector which detects whether a particular information file including menu generation information for the user interface screen depending on a device model exists;

a particular information extractor which extracts the menu generation information from the particular information file when the particular information file exists; and

a screen generator which generates the user interface screen based on the extracted menu generation information.

10. The apparatus of claim 9, wherein the screen generator generates the user interface screen based on predetermined default values when the particular information file does not exist.

11. A method of generating a user interface screen for a device driver, the method comprising:

installing the device driver and the user interface screen for the device driver in a host and requesting model dependent information from a device connected to the host;

receiving the model dependent information and model identification information from the device and determining whether the device can be driven by the device driver based on the model dependent information; and

in response to the determining that the device can be driven by the device driver, generating the user interface screen for the device driver using the model dependent information of the device.

12. The method of claim 11, wherein the generating the user interface screen comprises:

storing the model dependent information when the device can be driven by the device driver; and

generating the user interface screen using the model dependent information when generation of the user interface screen is requested.

13. The method of claim 12, wherein the storing the model dependent information comprises:

requesting model dependent information from the device when the device can be driven by the device driver and a user requests an update of the model dependent information; and

storing the model dependent information and the model identification information in the host.

14. An apparatus generating a user interface screen for a device driver, the apparatus comprising:

a driver dependent information requestor which requests model dependent information from a device connected to a host after the device driver and the user interface screen for the device driver are installed in the host;

an information input unit which receives the model dependent information and model identification information from the device;

a comparator which compares the model identification information with installed driver information; and

a screen generator which generates the user interface screen using the model dependent information when the model identification information is the same as the installed driver information,

wherein the installed driver information identifies a model of a supported device that can be driven by the installed device driver.

15. A method of generating a device driver of a peripheral, comprising:
selecting a predetermined number of functional components of the device driver;
modifying the selected predetermined number of functional components of the device driver while leaving other functional components of the device driver unmodified; and
updating the device driver with the modified predetermined number of functional components.

16. The method of claim 15, further comprising:
generating a user interface based on information extracted from the updated device driver.

17. The method of claim 16, wherein the generating the user interface is done in real time.

18. The method of claim 16, wherein the generating the user interface comprises reflecting the information extracted to a single user interface DLL file.

19. The method of claim 16, wherein the generating the user interface comprises modifying the user interface based on a model of the peripheral.

20. The method of claim 15, further comprising:
requesting and receiving model dependent information from the peripheral; and
generating a user interface based on information extracted from the updated device driver and the received model dependent information when the peripheral is compatible with the updated device driver.

21. A device driver generating apparatus of a peripheral, comprising:
an installation programmer which generates separate files corresponding to respective functional components of a device driver and selectively modifies only a file corresponding to a functional component to be modified to generate a device driver installation program;
a driver generator which generates the device driver based on functional components selected by a user in the device driver installation program;
a particular file extractor which extracts menu generation information from one of an information file which stores menu generation information corresponding to different models of the peripheral, and a default information file which stores generic menu generation information;
and
a screen generator which generates a user interface screen based on the extracted menu generation information.

22. The apparatus of claim 21, further comprising:
a file detector which selectively controls the particular file extractor to extract one of an information file which stores menu generation information corresponding to different models of the peripheral, and a default information file which stores generic menu generation information based on whether an information file including menu generation information for a user interface screen depending on a device model exists.

23. The apparatus of claim 21, wherein the screen generator further comprises:
a driver dependent information requestor which requests model dependent information from the peripheral;
an information input unit which receives the model dependent information and model identification information from the peripheral in response to the request for model dependent information; and
a comparator which compares the model identification information with the device driver, wherein the user interface screen is based on the extracted menu generation information and the received model dependent information when the peripheral is compatible with the device driver.

24. The apparatus of claim 21, wherein the peripheral is a printer.

25. The apparatus of claim 24, wherein the each of the functional components has a device mode (DEVMODE) data structure value as a parameter.

26. A computer readable recording medium recording a program that executes a method of generating a device driver of a peripheral, wherein the method comprises:
selecting a predetermined number of functional components of the device driver;
modifying the selected predetermined number of functional components of the device driver while leaving other functional components of the device driver unmodified; and
updating the device driver with the modified predetermined number of functional components.

27. A computer readable recording medium recording a program that executes a method of generating a user interface screen for a device driver, wherein the method comprises:
installing a device driver and a user interface for the device driver in a host and requesting model dependent information from a device connected to the host;
receiving the model dependent information and model identification information from the device and determining whether the device can be driven by the device driver based on the model dependent information; and
in response to determining that the device can be driven by the device driver, generating a user interface screen for the device driver using the model dependent information of the device.